Assignment 4: Neural Network Report

Annealing the Learning Rate

Having a constant small learning rate would take a long time for us to converge. Likewise, if we have a constant big learning rate, the gradient descent can overshoot the minimum, fail to converge or possible even diverge. Thus, I studied the effect of annealing the learning rate. I implemented two approaches, step decay (reduce learning rate by half every X epochs) and the search then converge schedule (converge after X initial epochs). Each case was run two times and the average errors were recorded. Each case kept all other factors constant (0.35 learning rate initially, k = 1 and 20 nodes in hidden layer).

	Step Decay Strategy						Search Then Converge					
# Classification Errors With Training Set After (out of 1863 examples)	X = 20	X = 50	X= 75	X = 100	X= 150	X= 200	X = 20	X = 50	X= 75	X = 100	X= 150	X= 200
1 Epochs	1854	1836.5	1843	1844.5	1847	1854	1681	1682.5	1691	1680	1690	1731.5
200 Epochs	75	78	92	96.5	90.5	90	8	7	3.5	4.5	1	3
400 Epochs	20	16	21	31	21.5	25	3	3	4	3.5	1	2.5
600 Epochs	9.5	6.5	7	9.5	10	7	2.5	2.5	3	3	1	2.5
800 Epochs	5.5	3.5	3.5	7	5.5	4	3	2.5	3	3	1.5	2.5
1000 Epochs	5.5	2	1.5	3	3.5	3.5	2.5	2.5	2	3	1	2.5
# Classification Errors Test Set (out of 960)	51	53.5	43.5	40	43	45	48.5	50	38	40	44	48.5

From these results, it is apparent that search then converge is superior to the step decay strategy. It consistently took fewer Epochs to get to acceptable levels of classification errors of the training set while maintaining low classification errors with the test set. The trials for each case sometimes varied significantly, and ideally more trials would be beneficial, but it appears as if Search the Converge with X = 75 is the optimal choice.

Introducing A Validation Set

After introducing a validation set, we generally never met the desired threshold for validation accuracy (97.5%) for us to break early. We consistently reach the 1000 Epoch upper bound limit though. Approximately 20% of the data was used for validation and 20% was used for testing. Achieving a 95% accuracy on the test data is common given the use of a validation set. Sometimes we are lucky and achieve 97%.

Number of Nodes in the Hidden Layer

Num Nodes	1	5	10	15	20	25	30	35	40	45
Training Data Error (%)	100	5.1	0.709	0.17	0.11	0.059	0.11	0.295	0.059	0.059

Assignment 4: Neural Network Report

Test Data Error (%)	83.71	9.38	6.90	4.60	4.24	4.24	4.07	4.24	2.83	3.89

While keeping all other factors constant, the number of nodes in the hidden layer was adjusted to different amounts. Each case was run two times and the average was recorded. Although a hidden layer with 25—35 nodes would suffice with a pretty good test data error rate, I found that 40 nodes consistently performed better (most trials had less than ~4% test data error rate). However, this came with a huge sacrifice in computation time, as the test took much longer with more nodes. This would have been even larger if we had more data. After further tests (36-39 and 41-44 nodes), 40 hidden nodes was the fewest number of nodes that performed consistently.

Sigmoid Function K Value

All other parameters were kept constant (0.35 base learning rate, 40 hidden nodes). Each case was run two times and the average was recorded.

K Value	0.5	1	1.5	2.0	2.5	3.0	3.5
Training Data Error (%)	0.059	0.177	0.177	0.059	0.11	0.47	11.1
Test Data Error (%)	3.18	3.53	2.81	3.71	4.96	5.13	18.58

Thus, from these results, we can conclude that K values greater than 2.0 are not that great. it is apparent that lower values for K are more optimal. In my very small test, it appears as if 1.5 is the best value.

Optimal Parameters

40 hidden nodes, K = 1.5 and search the converge with x = 75 seem to be the best values.

The Variability of the Data

One of the main limitations of my study were the number of trial runs I did for each case. I noticed huge variability in training data and test data error rates for any case. It all really depended on how lucky the division of the training, test and validation data was for each of the cases. To understand these parameters, it would be more ideal to use the average over 100 different trials for each case as opposed to just two (not feasible due to time constraints). Not only that, I didn't test all possible values in the continuous range for each of the parameters (for example, K = 1.6). It would be interesting to see how slight changes would affect the performance of the neural network.

Articles Used

https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1
https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10
http://cs231n.github.io/neural-networks-3/#anneal
https://www.willamette.edu/~gorr/classes/cs449/momrate.html